AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions and listings of claims in the application.

1.   (currently amended) A method for providing content, comprising the steps of:

receiving a request from a user device for particular content, the request is received at a server;

accessing a markup language description of the particular content, the markup language description includes markup language source code which describes a behavior of the particular content on a user interface of the user device based on user interactions with the particular content via the user interface, the markup language description includes a reference to a view instance element;

at the server, converting the markup language description to a script source code by:

accessing the markup language source code to locate a tag name and associated attributes for the view instance element, the tag name is parent tag name;

accessing the markup language source code to locate an associated child tag name and associated attributes in the view instance element; and

creating script source code based on the markup language source code, the script source code includes a script instruction to call an instantiation view function in a function call, the script instruction includes the child tag name and associated attributes in the function call as parameters; and

providing executable code for the user device by compiling the script source code to byte code, the step of compiling is performed at the server in response to the request; and

transmitting the executable code from the server to the user device for execution by the user device to allow a user to access the particular content via the user interface according to the behavior and the user interactions, the instantiation view function is called at a runtime of the executable code at the user device, and creates objects which are displayed on the user interface, based on the child tag name and associated attributes.

2.   (cancelled)

- 2 -

3.   (cancelled)

4.   (previously presented) A method according to claim 1, wherein:

the user device includes a rendering entity and a browser, the rendering entity is a plug-in to the browser, the plug-in is embedded in the browser before the request, and the rendering entity executes the executable code.

5.   (previously presented) A method according to claim 1, wherein:

the script source code comprises ActionScript and the byte code comprises corresponding ActionScript byte code.

6.   (cancelled)

7.   (previously presented) A method according to claim 1, wherein:

the request for particular content is received from a browser in which a rendering entity is present as a plug-in to the browser, the browser is at the user device, and the rendering entity executes the executable code.

8.   (previously presented) A method according to claim 1, wherein:

the particular content comprises a media file comprising video;

the markup language source code includes a source attribute which references a name of the media file, the source attribute is within a window tag in the markup language source code; and

the media file is transmitted with the executable code from the server to the user device for use by a rendering entity at the user device in allowing the user to access the video in a window on the user interface when the media file is referenced by the source attribute within the window tag when the executable code is executed.

9.   (cancelled)

10.   (cancelled)

11.   (previously presented) A method according to claim 1, further comprising the step of:

authenticating the request, the steps of compiling and transmitting are only performed if the step of authenticating is successful, and different types of authenticating are provided for different types of content, including content types of application, data and service.

12.   (cancelled)

13.   (previously presented) A method according to claim 1, wherein the request from the user device for the particular content is a request for a first application, the first application runs on the user device when the executable code is executed at the user device, the method further comprising the steps of:

receiving a request at the server from the first application running on the user device for a second application;

in response to the request, accessing and compiling the second application; and

transmitting the compiled second application from the server to the user device.

14.-51.   (cancelled)

52.   (previously presented) A method according to claim 4, wherein:

the server has separate object code generators and compilers for different types of rendering entities;

the request is received at the server from the user device and includes an indication that identifies a type of the rendering entity from among the different types of rendering entities; and

the executable code is created specific for the type of rendering entity using corresponding ones of the object code generators and compilers, in response to the indication.

53.   (previously presented) A method according to claim 1, wherein:

- 4 -

the executable code comprises one or more binary files.

54.   (previously presented) A method according to claim 1, wherein:
the executable code comprises object code.

55.-68.   (cancelled)

69.   (previously presented) A method according to claim 1, wherein:
the compiling comprises parsing the markup language description to identify first and second types of elements in the markup language description, providing the first type of element to a first compiling module which is appropriate for the first type of element to obtain first object code, providing the second type of element to a second compiling module which is appropriate for the second type of element to obtain second object code, and assembling the first and second object code into a single executable; and
the transmitting the executable code comprises transmitting the single executable to the user device.

70.   (previously presented) A method according to claim 69, wherein:
the first type of element provides a script which defines the behavior of the particular content, and the second type of element defines a connection to an external data source.

71.   (cancelled)

72.   (cancelled)

73.   (previously presented) A method according to claim 4, wherein:
the rendering entity is a Flash player.

74.-80. (cancelled)

81.   (previously presented)   A method according to claim 8, further comprising:

providing an object in the executable code which includes fields storing attributes which identify the name and a format of the media file, the name and format are provided via the user interface when the media file is rendered.

82.   (previously presented)   A method according to claim 8, wherein:

the request for particular content is received from a browser in which a plug-in to the browser is present, the browser is at the user device, and the plug-in renders the media file.

83.   (previously presented)   A method according to claim 1, wherein:

the executable code provides a script based on the script source code which is executed when a specified event occurs when a user interacts with the particular content via the user interface, the specified event is based on at least one of user control of a pointing device and a key press.

84.   (cancelled)

85.   (previously presented)   A method according to claim 1, wherein:

the instantiation view function is predefined, and the executable code, when executed at the user device, calls the predefined instantiation view function associated, and passes the attributes to the predefined instantiation view function.

86.   (previously presented)   A method according to claim 1, wherein:

the instantiation view function is predefined, and the executable code, when executed at the user device, calls the user-defined instantiation view function associated, and passes the attributes to the user-defined instantiation view function.

87.   (previously presented)   A method according to claim 1, wherein the particular content includes media content, the method further comprising:

receiving the media content at the server, from an external data source;

creating an object representation of the media content, the object representation includes the

- 6 -

media content and fields which store attributes of the media content, the attributes include a name of the media content and a format of the media content;

removing the media content from the object representation and inserting a reference to the media content into the object representation in place of the media content, then compiling the object representation to byte code;

creating a tag header;

adding the media content, but not the compiled object representation, to the tag header; and

transmitting the compiled object representation with the compiled markup language description from the server to the user device for execution by the user device to provide the particular content and the data via the user interface according to the behavior and the user interactions.

88.   (previously presented) A method according to claim 87, further comprising:

assembling the compiled markup language description and the compiled object representation into a single executable which is transmitted as the executable code from the server to the user device.

89.   (previously presented) A method according to claim 87, further comprising:

transcoding the media content before the adding the media content to the tag header, the transcoding is separate from the compiling of the object, the media content comprises audio which is transcoded between MP3 and WAV.

90.   (previously presented) A method according to claim 87, further comprising:

transcoding the media content before the adding the media content to the tag header, the transcoding is separate from the compiling of the object, the media content comprises video which is transcoded between any two of the following formats: MPEG, MPEG2, SORENSON, REAL, and Animated GIF.

91.   (previously presented) A method according to claim 87, further comprising:

transcoding the media content before the adding the media content to the tag header, the

transcoding is separate from the compiling of the object, the media content comprises an image which is transcoded between any two of the following formats: JPEG, GIF, BMP, and PNG.

92.   (previously presented) A method according to claim 87, further comprising:

transcoding the media content before the adding the media content to the tag header, the transcoding is separate from the compiling of the object, the media content comprises graphics which is transcoded between any two of the following formats: SVG, HTML, PDF and SWF.

93.   (previously presented) A method according to claim 87, wherein:

the media content comprises audio.

94.   (previously presented) A method according to claim 87, wherein:

the media content comprises video.

95.   (previously presented) A method according to claim 87, wherein:

the media content comprises a movie.

96.   (previously presented) A method according to claim 87, wherein:

the media content comprises an animation.

97.   (previously presented) A method according to claim 87, wherein:

the media content comprises a .SWF file.

98.-102   (cancelled)

103.   (previously presented) A method according to claim 1, wherein:

the objects which are created by the instantiation view function, and displayed on the user interface, include at least one of windows, dialogs, buttons, images, text fields, banners and animation.

104.   (previously presented)   A method according to claim 1, wherein:

the objects which are created by the instantiation view function, and displayed on the user interface, include at least one of dialogs, buttons, and text fields.

105.   (previously presented)   A method according to claim 1, wherein:

the instantiation view function, when called at the runtime of the executable code at the user device, creates a sound object on the user device, based on the tag name and associated attributes.

106.   (cancelled)

107.   (currently amended)   A method according to claim 1, 106, wherein:

the instantiation view function is called recursively at the runtime of the executable code at the user device, first with the parent tag and associated attributes and then with the child tag and associated attributes, and results from calling the instantiation view function with the child tag and associated attributes are attached to results from calling the instantiation view function with the parent tag and associated attributes.

108.   (previously presented)   A method according to claim 1, wherein:

when the instantiation view function is called at the runtime, a class which is associated with the tag name is identified, and a separate table of instantiation functions for the class, indexed by tag name, is accessed, to create the objects which are displayed on the user interface.

109.   (previously presented)   A method according to claim 1, wherein:

when the instantiation view function is called at the runtime, a table of instantiation functions, indexed by tag name, is accessed, to create the objects which are displayed on the user interface.

110.   (previously presented)   A method according to claim 1, wherein:

the view instance element includes a reference to data for rendering on the user interface, and the markup language description defines a connection to an external data source for the data, the

external data source is external to the server; and

the server accesses the data at the external data source based on the markup language description.

111.-113.   (cancelled)

114.   (new) One or more processor readable storage devices having processor readable code embodied on the processor readable storage devices, the processor readable code for programming one or more processors to perform the method of claim 1.